Rapid-response Adaptive Computing Environments for LHC Physics Analysis

The University of Wisconsin-Madison High Energy Physics and Computer Science groups proposed to further strengthen this productive collaboration and work as one interdisciplinary team to define and build a *Rapid-response Adaptive Computing Environment* (RACE) for enabling timely extraction of LHC physics. In particular, they propose to build a RACE for CMS trigger system data validation and data forensics, in online data acquisition and offline data processing stages of CMS computing.

RACE project team worked on resource and knowledge management issues essentially independently in the first year. The resource management issues were studied by D.Bradley and S.Dasu from the Physics department, and M.Livny, S.Murphy, E.Paulson and A.Roy from the Computer Science department. The knowledge management issues were studied by B-C.Chen, L.Chen and R.Ramakrishnan from the CS department, and D.Bradley and S.Dasu from the Physics department.

Resource Management

The proposed first year deliverable for the resource management portion of the project was: RACE.v1: A prototype software package for creating *RACEs* for CMS at UW, for production and analysis, using Condor technologies. The first version includes only resource management capabilities.

The milestone that we expected to meet by August 2005 was: Completion of RACE.v1

The goals that we setup for RACE.v1 was to provide an ability to takeover a set of computational resources with high priority jobs temporarily, while not killing the running lower priority jobs on those resources. Condor was able to provide preemption of specially compiled, so called Standard Universe jobs, to maximize resource utilization. However, both CMS and ATLAS are not able to build Standard Universe jobs, which can be preempted and rerun on another machine when available. Therefore, they were losing large computing time due to higher priority job interrupts. The primary purpose of RACE.v1 is to circumvent this problem.

Solution 1: S. Murphy and E. Paulson developed and deployed a system consisting of multiple virtual machines (VMs) on the compute nodes of the Grid Laboratory of Wisconsin (GLOW), which is a shared facility with several campus sites. There are four VMs deployed on each GLOW compute node with two CPUs. At any time at most two VMs are active, and the other two VMs with lower priority jobs are suspended. On GLOW, jobs belonging to one of the groups (ATLAS experiment at present) are setup to occupy all sites with low priority jobs. These jobs are normally suspended on all sites, which are not owned by ATLAS, because they have their own jobs with higher priority using the resources. However, whenever there is no higher priority activity the background jobs use the resources. The net effect is that resource utilization on GLOW is maximized, and ATLAS experiment, which is running many simulation jobs, is able to exploit idling resources without specially compiled Standard Universe jobs.

Solution 2: D. Bradley had earlier implemented with CMS core software framework group the ability to restart jobs from where stopped. He augmented the CMS job management scripts to take advantage of these framework changes and save bookkeeping information so that restarted jobs can be monitored properly. These changes to CMS jobs, allow them to be run on opportunistic resources with greater efficiency.

Solution 3: D. Bradley implemented a new job submission agent for Jug, a job management software package developed in a previous ITR grant. The new submission agent executes the jobs through Condor's "Computing On Demand" (COD) system, rather than the normal batch queue. Using this system, we can submit high priority jobs through COD, and immediately take over machines that have been configured to allow us to run COD jobs. Any existing jobs on those machines are suspended while the COD job runs. There is no accounting or negotiation at this time, so the applicability to a system like GLOW where we need to define a formal COD usage policy will require some further COD development. We are finding out how much of this COD development overlaps with the MIT group's requests to the Condor team for PROOF related functionality.

Our present plans call for proceeding forward with generalizations of solution 1, so that it is applicable to multiple users, and completion of solution 3 by August 2005 to meet our milestone. These additions to Condor will form the first RACE.v1 package.

Knowledge Management

The proposed first year deliverable for the knowledge management portion of the project was: A framework developed in collaboration with physicists in CMS that is capable of using machine learning techniques to identify problems in streaming data.

Lei Chen began work on adapting and developing change detection and root cause identification algorithms under the supervision of Prof. R. Ramakrishnan. The physics team is interested in developing automated validation and fault identification mechanisms for its online trigger systems. D. Bradley and S. Dasu provided simulated CMS trigger data in which abnormal changes were deliberately introduced by varying input parameters. Lei Chen used those data sets to characterize the algorithms that he developed. The results were discussed in fortnightly meetings with participation from both groups. In summary, the algorithms that used statistical measures calculated on an aggregation of time series data provided the best results. This research is described in more detail below:

Change Detection and Root Cause Identification:

We have developed an automatic mechanism to continuously monitor the output stream of a complex system and to identify the problematic change of the internal control parameters. Our basic approach is a combination of supervised learning and model ensembles. A prediction model is constructed over a system simulator (developed by Bradley and Dasu) which can be tuned to provide both the positive and negative samples. The trained predictive classifier can then estimate the values of underlying control parameters by observing the events produced by the system. Based on the result of prediction, appropriate actions can be taken. For example, if the estimate value deviates too much from the expected one, an alarm could be sent to the system administrator indicating a potential problem.

In order to improve the accuracy of the prediction and reduce the number of false alarms, we are investigating ensemble methods, which apply prediction models for various chunks of the stream and make the prediction by summarizing the results from individual classifiers. We propose two types of ensembles: The first ensemble approach makes prediction over individual events in the output stream and aggregates the prediction result across a fixed time duration. The second approach first generates data summaries for nearby events and makes prediction based on these data summaries. Our preliminary results show that the ensemble approach can significantly reduce the false positives at the cost of a slight delay between the time when the system goes wrong and the time when the alarm is sent.

We are currently preparing a paper summarizing the techniques for single component control parameter monitoring.

Next Steps and Additional Related Work:

In the future, we want to adopt various techniques from the field of time series analysis to improve the detection accuracy. As one promising direction, the Hidden Markov Model (HMM) can be used to improve the accuracy of the detection by taking into account domain background knowledge such as the general likelihood for a control parameter to go wrong and the correlation between two control parameters. The other research direction is to not only identify the problematic control parameter but also quantify the degree of deviation. We also want to construct prediction models which use not only the value of the events in the data stream but also the deviation of the values to make the prediction.

Furthermore, we want to extend the error detecting techniques to systems which consist of multiple components. In such systems, each component is controlled by a different parameter and has its own output stream. The output of one component might be fed into the other component as the input. All the components in the system form a complex data flow network. The prediction of the state of one component should be made not only on the basis of its own output stream, but also the streams from other components which are connected with it. We want to use tools for multi-element analysis such as Bayes network to tackle this problem.

The physics group, that includes graduate students W. C. Hogg and J. Leonard, has begun work on developing trigger emulator software based on the new CMS software framework. The emulator results and the actual data from the experiment will be compared in online data quality monitoring software. Automatic machine learning based fault-finding algorithms evaluated by our group will be implemented in this program.

The CS group has also initiated research on how to find interesting subsets of data. Specifically, they proposed using the data partitioning induced by a multidimensional data model to identifying potentially interesting data subsets. They examined this in depth for specific notions of interesting subsets as characterized by the predictive models trained on these subsets, in a paper presented at the VLDB 2005 conference. We are exploring how we can exploit these ideas for LHC physics problem domain.